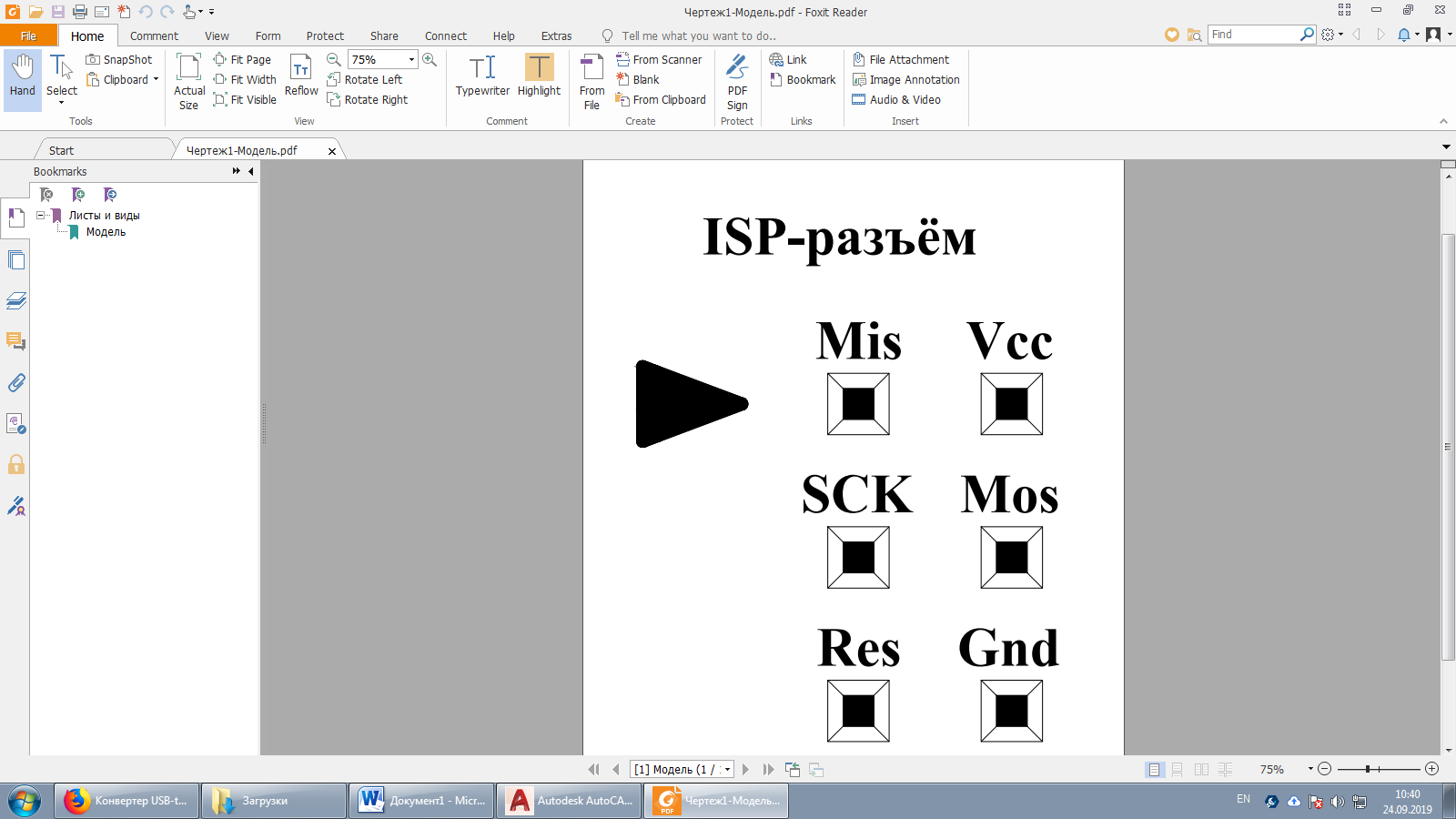
**Прошивка контроллеров Atmega.**

На базе данных контроллеров разработаны устройства УБД и датчик удара. Для программирования этих контроллеров требуется следующий инструментарий:

* программатор для контроллеров AVR. Например, USBASP. Распиновка разъёма данного программатора, это стандартный ISP разъём:

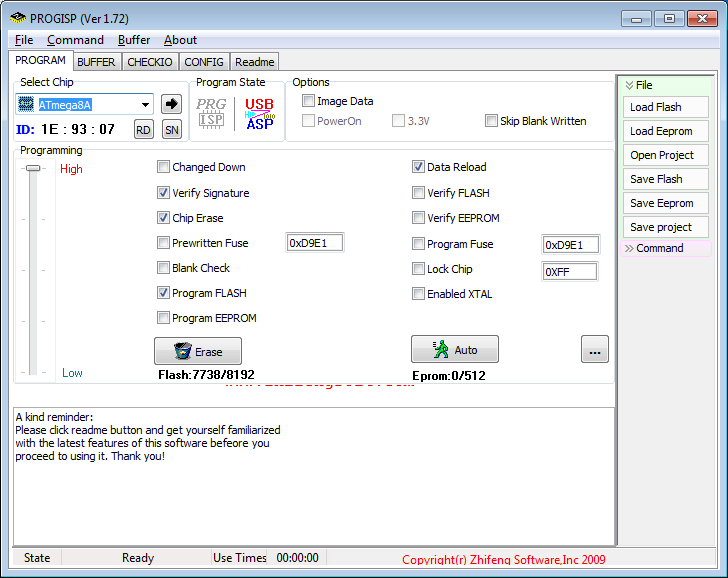


* переходник USB-UART. Например, на базе микросхемы ch340.
* программа для редактирования и компилирования программы контроллера BASCOM
* программа для редактирования и компилирования программы контроллера Atmel Studio
* программа для загрузки программы в контроллер ProgISP (стабильно работает 1.72 версия)
* программа для получения данных из серийного порта SerialMonitor (или любая аналогичная)
* программа для просмотра и редактирования схем и печатных плат проектов Kicad 5.1.4.1(не ниже)
* в некоторых случаях понадобится Arduino IDE, для редактирования и загрузки программ в Arduino



Для прошивки контроллера следует выбрать напряжение питания на программаторе, с помощью перемычки, 5В или 3.3В. Для того, что бы удостовериться в выборе напряжения, следует померить его на разъёме программатора. На рисунке показан общий вид программатора и перемычка выбора напряжения питания программируемого устройства.

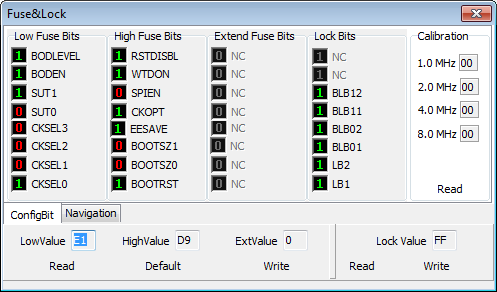
Далее подключив разъём программатора к ISP разъёму устройства требуется запустить программу ProgISP. Общий внешний вид программы и необходимые настройки показаны на рисунке ниже.

После запуска программы требуется выбрать программируемый контроллер из списка в поле “Select Chip”. Для датчика удара это Atmega8 или Atmega8A. Для УБД это Atmega128 или Atmega128A.

Далее требуется проверить правильность подключения программатора и работоспособность контроллера. Для этого достаточно считать его настройки (FUSES или фьюзы). Делается это следующим образом:

Фьюзы

Требуется зайти в окно настройки фьюзов нажатием на соответствующее поле.

 В открывшемся окне (Fuse&Lock) нажатием кнопки Read будут прочитаны настройки контроллера и подставлены в поля LowValue и HighValue. Возможные проблемы:

**Ошибка** “**Programmer can not be found.**” Говорит о том, что программатор либо не подключен, либо завис. Решается это подключением программатора или переподключением его в разъём USB.

**Ошибка** “**Chip Enable Program Error**” говорит о том, что контроллер не может быть прочитан. Причин этому может быть множество. Вот некоторые из них:

Контроллер плохо припаян к плате; на контроллер не приходит питание; в контролер загружены не корректные фьюзы (можно сразу менять контроллер); не правильно подключен разъём программатора; контроллер выведен из строя…….. и тд.

Также при множественном чтении настроек в поля LowValue и HighValue могут заноситься постоянно разные данные, это говорит о плохом питающем напряжении, к примеру, на контроллер не приходит линия Gnd или Vcc.

Если есть возможность несколько раз стабильно прочитать фьюзы из контроллера это говорит о том, что он исправен и готов к прошивке.

Для этого сначала требуется настроить эти самые фьюзы с помощью полей LowValue и HighValue. [Датчик удара: LowValue=E4 HighValue=D9] [УБД: LowValue=FF HighValue=C1 ExtValue=FF]. После заполнения данных полей нажать на кнопку “Write”. Если всё прошло без ошибок, то для контроля можно несколько раз прочитать фьюзы. После закрываем окно Fuse&Lock.

Далее требуется выбрать hex-файл скомпилированной программы. Это делается нажатием кнопки “Load Flash”, и в открывшемся окне указать путь до hex-файла.

**Датчик удара** […\UBD\Программы\Датчик удара\Accel\_LIS302\LIS302DL.hex]

**УБД** […\UBD\Программы\Программа МК\WiFi.hex]

Чтобы загрузить hex-файл в контроллер нажимаем кнопку “Auto”, после завершения загрузки контроллер автоматически начнёт работу.

**Редактирование программы для УБД**

Проект для УБД можно открыть и редактировать средой BASCOM IDE.

Запустив данную среду программирования, следует открыть файл проекта. Файл-Открыть в окне появившемся окне указать тип файлов “Project files (\*.prj)”. Путь …\UBD\Программы\Программа МК\UBD.prj.

В данном проекте, для настойки работоспособности устройства требуется настроить следующие константы и переменные:

Plata версия платы. 1-старая плата, 2-новая плата [WiFi]

Debug\_info вывод отладочной информации в UART [WiFi]

Print\_info вывод информации о состоянии устройства в UART [WiFi]

Wifi номер дампа для сетевых настроек [Vars]

Ud\_eep если 0-не сохранять удар в eeprom, если 1-сохранять удар в eeprom [Vars]

Ud\_on если 0-удар отключен, если 1-удар включен [Vars]

Ig\_ud\_off отключение зажигания при ударе 0-не отключается, 1-отключается [Vars]

Дампы сетевых настроек. Имеются три дампа в файле Data:

Ip\_adr дамп IP-адресов сервера

Wifi\_ap дамп имён точек доступа

Wifi\_pass дамп паролей от точек доступа

Нумерация записей в дампе начинаются с нуля и выбираются с помощью переменной Wifi. К примеру, если задать данной переменной значение 2, то IP-сервера будет 192.168.0.30, имя точки доступа Kslift, пароль от точки доступа A0C4B096D4.

Ip\_adr: IP-адреса сервера [Data]

Data \_

"10.11.22.50", \_

"10.0.0.3", \_

"192.168.0.30", \_

"192.168.1.20",

Wifi\_ap: Имена точек доступа

Data \_

"MJR-Sklad", \_

"NETGEAR", \_

"Kslift", \_

"dimaib"

Wifi\_pass: Пароли точек доступа

Data \_

"DabuDaDe", \_

"111222333444", \_

"A0C4B096D4", \_

"230988DVG"

После всех настроек сделанных в прошивке следует нажать клавишу F7, что запустит компиляцию. После удачной компиляции программу следует загрузить в контроллер.